

Тема 8. Множества (set, frozenset)

Множеством называется совокупность некоторых элементов, объединенных каким-либо общим признаком. Элементами множества могут быть числа, фигуры, предметы, понятия и т. п.

С точки зрения программирования, множество – это контейнер, содержащий **неповторяющиеся** элементы в случайном порядке.

8.1. Создание множеств

Множество можно задать разными способами.

1. Пустое множество.

```
a = set()
print(a)
```

```
set()
```

2. Множество из набора символов.

```
a = set('Python')
print(a)
```

```
{'h', 'P', 'o', 't', 'n', 'y'}
```

3. Пустое множество нельзя создать при помощи литерала, т. е. запись типа `a={}` создаст не множество, а словарь.

4. Задаем элементы вручную.

```
a = {'a', 'b', 'c'}
print(a)
```

```
{'c', 'a', 'b'}
```

5. Генерирование множеств.

```
a = {i*2 for i in range(5)}
print(a)
```

```
{0, 8, 2, 4, 6}
```

6. Ввод элементов множества с клавиатуры.

Способ 1:

```
a = {input('Введите элемент множества: ') for i in range(5)}
print(a)
```

Способ 2:

```
a = set(input('Введите элемент множества: ') for i in range(5))
print(a)
```

При создании множеств нужно учитывать, что элементы в множестве повторяться не могут. Если мы создадим множество и впишем в него несколько повторяющихся элементов, то в нем останется каждый элемент в одном экземпляре.

```
a = set('programming')
print(a)
```

```
{'p', 'a', 'i', 'n', 'g', 'o', 'm', 'r'}
```

8.2. Операции над множествами

Прежде чем разбирать операции над множествами, приведем несколько определений.

Определение 1. Равенство множеств

Множества называются равными, если они состоят из одних и тех же элементов.

Определение 2. Понятие подмножества (принадлежность множеству)

Множество В называется подмножеством множества А (принадлежит множеству А) тогда и только тогда, когда каждый элемент множества В содержится в множестве А.

Определение 3. Объединение множеств

Объединением множеств А и В называется новое множество, состоящее из всех элементов множества А и множества В, исключая повторяющиеся.

Определение 4. Пересечение множеств

Пересечением двух множеств называется новое множество, состоящее из элементов, одновременно принадлежащих и первому и второму множествам.

Определение 5. Разность множеств

Разностью двух множеств называется новое множество, элементы которого принадлежат первому множеству, но не принадлежат второму множеству.

Операции, не изменяющие множества

Оператор, метод	Описание	Пример
len(a)	Число элементов в множестве (размер множества).	a={1,3,5,7,8} print(len(a)) Получим: 5
n in a	Проверяет принадлежность n множеству a. Если принадлежит, то возвращает True, иначе False.	a={1,3,5,7,8} print(3 in a) Получим: True
a == b	Проверяет равенство множеств.	a={1,3,4} b={3,1,4} print(a==b)

		Получим: True	
a.issubset(b) или a<=b	Проверяет принадлежность элементов множества b множеству a . Если a является подмножеством, то возвращает True, иначе False.	a={1,3,4} b={3,1} print(b.issubset(a))	a={1,3,4} b={3,1} print(a>=b)
a.union(b,c,...) или a b c ...	Объединение нескольких множеств.	a={1,3,4} b={3,1,5} print(a.union(b))	a={1,3,4} b={3,1,5} print(a b)
a.intersection(b,c,...) или a & b & c & ...	Пересечение нескольких множеств.	a={1,3,4} b={3,1,5} print(a.intersection(b))	a={1,3,4} b={3,1,5} print(a & b)
a.difference(b,c,...) или a-b-c-...	Элементы множества a , не принадлежащие ни одному из множеств в скобках. Вычитание из множества a .	a={1,3} b={3,1,4,5} print(b.difference(a))	a={1,3} b={3,1,4,5} print(b-a)
a.symmetric_difference(b) или a^b	Элементы, принадлежащие только одному из множеств a или b .	a={1,3,0} b={3,1,4,5} print(b.symmetric_difference(a))	a={1,3,0} b={3,1,4,5} print(b^a)
b = a.copy() или b = a	Создание копии множества.	a={1,3,0} b=a.copy() c=a print(b, end=' ') print(c)	
		Получим: {0, 1, 3} {0, 1, 3}	

Операции, изменяющие множества

Оператор, метод	Описание	Пример	
a.update(b,c,...) или a = b = c = ...	Объединение множеств, результат записывается в первое множество.	a={1,3,0} b={2,5} a.update(b) print(a)	a={1,3,0} b={2,5} a = b print(a)
a.intersection_update(b,c,...) или a &= b &= c &= ...	Пересечение множеств, результат записывается в первое множество.	a={1,3,2} b={2,5} a.intersection_update(b) print(a)	a={1,3,2} b={2,5} a &= b print(a)

		Получим: {2}	
a.difference_update(b,c,...) или a -= b -= c -= ...	Вычитание из множества множеств в скобках, результат записывается в первое множество.	a={1,3,2} b={2,5} a.difference_update(b) print(a) Получим: {1, 3}	a={1,3,2} b={2,5} a -= b print(a)
a.symmetric_difference_update(b) или a ^= b	Элементы, принадлежащие только одному из множеств, a или b , результат записывается в множество a .	a={1,3,2} b={2,5} a.symmetric_difference_update(b) print(a) Получим: {1, 3, 5}	a={1,3,2} b={2,5} a ^= b print(a)
a.add(b)	Добавляет элемент b в множество a .	a={1,3,2} a.add(5) print(a) Получим: {1, 2, 3, 5}	
a.remove(b)	Удаляет элемент b из множества a . Если такого элемента нет, то ошибка: KeyError.	a={1,3,2} a.remove(3) print(a) Получим: {1, 2}	
a.discard(b)	Удаляет элемент b из множества a , не вызывая ошибку при отсутствии такого элемента.	a={1,3,2} a.discard(3) print(a) Получим: {1, 2}	
a.pop()	Удаляет первый элемент из множества. Так как множества не упорядочены, нельзя точно сказать, какой элемент будет первым.	a={4,3,2} a.pop() print(a) Получим: {3, 4}	
a.clear()	Очищает множество от элементов.	a={4,3,2} a.clear() print(a) Получим: set()	

8.3. Перебор элементов множества

Прежде чем разбирать операции над множествами, приведем несколько определений.

```
a = {3, 5, 7, 2, 6}
for i in a:
    print(i, end=' ')
```

```
2 3 5 6 7
```

Этот способ позволяет только вывести элементы и затем их обработать. Изменить элемент множества таким образом не получится.

8.4. Неизменяемые множества

Неизменяемые множества **frozenset** представляют собой те же самые множества **set**, за исключением того, что **set** – это изменяемый тип данных, а **frozenset** – неизменяемый тип данных. Подобная ситуация была со списками и кортежами, где списки – изменяемый тип данных, а кортежи – неизменяемый.

Неизменяемые множества создаются при помощи функции **frozenset()** из любого итерируемого объекта. Причем множества, созданные из одного и того же итерируемого объекта при помощи **set()** и **frozenset()**, будут совпадать, но типы данных все же будут разные.

```
a = set('Python')
b = frozenset('Python')
print(a == b)
print(type(a))
print(type(b))
```

Результат:

```
True
<class 'set'>
<class 'frozenset'>
```

Над неизменяемыми множествами **frozenset** определены все функции множеств **set**, которые не изменяют непосредственно само множество. Если попытаться добавить в созданное неизменяемое множество какой-либо элемент, в результате будет выдана ошибка.